"Exploring Machine Learning Models for Data Classification"

MNIST Databas



Maya Sobh

EC Utbildning

Machine Learning – Kunskapskontrollen2

202403

# Abstract

**En kort sammanfattning över ditt arbete och de viktigaste resultaten skrivet på engelska, cirka 5 meningar totalt.**

In this work, we utilized the MNIST dataset to evaluate various machine learning models for digit recognition. We started by preprocessing the data and then trained several classifiers, including Linear Support Vector Classifier, Random Forest, and Extra Trees. Performance evaluation on a validation set revealed Random Forest and Extra Trees as top performers. Confusion matrices provided insights into the models' classification capabilities, with Random Forest exhibiting robust performance. Subsequent testing on an independent test set reaffirmed the superiority of the Random Forest classifier, achieving an accuracy of 93.6%. This study underscores the effectiveness of ensemble methods for digit recognition tasks and highlights the importance of comprehensive model evaluation for reliable performance assessment in machine learning applications.

# Förkortningar och Begrepp

Abbreviations and Concepts Simplified:

1. LinearSVC : Linear Support Vector Classifier.

2. Random Forest Classifier : Ensemble of decision trees.

3. Extra Trees Classifier : Variation of Random Forest with random splits.

4. MNIST : Handwritten digit dataset.

**Skapas automatiskt i Word genom att gå till Referenser > Innehållsförteckning.**

Innehållsförteckning

[Abstract 2](#_Toc160654338)

[Förkortningar och Begrepp Detta avsnitt behövs oftast inte. 3](#_Toc160654339)

[1 Inledning 1](#_Toc160654340)

[1.1 Underrubrik – Exempel 1](#_Toc160654341)

[2 Teori 2](#_Toc160654342)

[2.1 Exempel: LinearSVC 2](#_Toc160654343)

[2.1.1 Exempel:Random Forest 2](#_Toc160654344)

[2.1.2 Exempel: ExtaTrees 2](#_Toc160654345)

[3 Metod 3](#_Toc160654348)

[4 Resultat och Diskussion 4](#_Toc160654349)

[5 Slutsatser 5](#_Toc160654350)

[6 Teoretiska frågor 6](#_Toc160654351)

[7 Självutvärdering 7](#_Toc160654352)

# Inledning

Syftet med detta arbete är att utforska igenkänningsalgoritmernas påverkan i vardagen, deras positiva effekter och potentiella risker. Genom att genomföra ett nummerigenkänningsprojekt på tre olika modeller strävar vi efter att identifiera den mest effektiva. Vi kommer att svara på följande frågor:

1. Vilken modell är den mest effektiva?

2. Vilka fördelar ger användningen av denna modell i vardagen?

3. Vilka risker kan vara förknippade med dess användning?

## Underrubrik – Exempel

MNIST-databasen är en välkänd samling handskrivna siffror som ofta används för att utvärdera och testa maskininlärningsalgoritmer, särskilt för bildigenkänning. Databasen innehåller ursprungligen 70 000 svartvita bilder på handskrivna siffror (0 till 9) och är uppdelad i en träningsuppsättning med 60 000 bilder och en testuppsättning med 10 000 bilder. Varje bild är av storleken 28x28 pixlar. MNIST-databasen är populär inom maskininlärningsgemenskapen på grund av sin tillgänglighet, standardisering och relativa enkelhet, vilket gör den till ett vanligt benchmark för att jämföra prestanda hos olika maskininlärningsalgoritmer för bildigenkänning.

# Teori

## Exempel: LinearSVC

Linear Support Vector Classification.

Similar to SVC with parameter kernel=’linear’, but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples.

The main differences between **[LinearSVC](https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html" \l "sklearn.svm.LinearSVC" \o "sklearn.svm.LinearSVC)** and [**SVC**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC) lie in the loss function used by default, and in the handling of intercept regularization between those two implementations.

This class supports both dense and sparse input and the multiclass support is handled according to a one-vs-the-rest scheme.

https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

### Exempel: Random Forest

A random forest classifier.

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. Trees in the forest use the best split strategy, i.e. equivalent to passing splitter="best" to the underlying **[DecisionTreeRegressor](https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html" \l "sklearn.tree.DecisionTreeRegressor" \o "sklearn.tree.DecisionTreeRegressor)**. The sub-sample size is controlled with the max\_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree.

For a comparison between tree-based ensemble models see the example [Comparing Random Forests and Histogram Gradient Boosting models](https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_hist_grad_boosting_comparison.html#sphx-glr-auto-examples-ensemble-plot-forest-hist-grad-boosting-comparison-py).

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

### Exempel: ExtraTree

An extra-trees classifier.

This class implements a meta estimator that fits a number of randomized decision trees (a.k.a. extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

# Metod

**Random Forest** Den slumpmässiga skogen består av flera beslutsträd, som är oberoende tränade på delmängder av inlärningsdataset (packningsmetod). Var och en producerar en uppskattning, vilket är den uppsättning resultat som ger den slutliga förutsägelsen som ger minst varians. Det handlar kort och gott om att ta inspiration från de olika åsikterna, och ta itu med samma problem, för att förstå det bättre. Varje modell är slumpmässigt fördelad i delmängder av beslutsträd

**Extra Trees** är en variant av Random Forest-algoritmen som använder ännu mer slumpmässighet vid skapandet av beslutsträd. Det gör nodsplitningar med slumpmässiga delmängder av både träningsdata och funktioner. Detta minskar överanpassning och kan förbättra prestanda, särskilt med högdimensionella data.

**LinearSVC** är en algoritm för maskininlärning som används för binär klassificering och använder en linjär beslutningsgräns för att separera datapunkter i olika klasser. SVC står för Support Vector Classifier, och "Linear" indikerar att den använder en linjär funktion för att definiera klassgränserna. Den strävar efter att hitta den optimala linjen som separerar datapunkterna med maximal marginal.

# Resultat och Diskussion

|  |  |
| --- | --- |
| **RMSE för olika modeller** | |
| Random Forest | 0,939 |
| ExtraTree | 0,947 |
| LinearSVC | 0.859 |

Tabell 1: Root Mean Squared Error (RMSE) för de fyra valda modellerna.

En bild som visar text, Teckensnitt, skärmbild, linje

Automatiskt genererad beskrivning

Figur 1: Hur man lägger in tabell eller figur nummer samt beskrivning.

# Slutsatser

"In this study, we conducted an investigation into recognition algorithms' impact in everyday scenarios, emphasizing their positive contributions and potential risks. Employing three different models—Random Forest, Extra Trees, and Linear Support Vector Classifier (LinearSVC)—we aimed to determine the most effective one. We addressed key questions regarding effectiveness, advantages, and risks associated with each model. Our methodology included model instantiation, individual training on provided data, and subsequent evaluation of their performance on validation data. Based on our findings, the Extra Trees model emerged as the most effective, showcasing robust performance in digit recognition tasks. While highlighting its advantages in real-world applications, we also acknowledged potential risks. This conclusion encapsulates our study's outcomes."

# Teoretiska frågor

1. Kalle delar upp sin data i ”Träning”, ”Validering” och ”Test”, vad används respektive del för?

Kalle delar upp sin data i tre delar för olika ändamål:

* Träning: Används för att träna modellen eller algoritmen genom att anpassa dess parametrar till datan.
* Validering: Används för att utvärdera modellens prestanda och justera eventuella hyperparametrar.
* Test: Används för att slutgiltigt utvärdera modellens prestanda på oberoende data. Det ger en uppfattning om hur väl modellen generaliserar till nya exempel.

1. Julia delar upp sin data i träning och test. På träningsdatan så tränar hon tre modeller; ”Linjär Regression”, ”Lasso regression” och en ”Random Forest modell”. Hur skall hon välja vilken av de tre modellerna hon skall fortsätta använda när hon inte skapat ett explicit ”valideringsdataset”?

När Julia inte har ett explicit valideringsdataset bör hon använda **korsvalidering** för att utvärdera prestandan hos sina tre modeller: **Linjär Regression**, **Lasso Regression** och **Random Forest**. Korsvalidering innebär att man delar upp träningsdatan i flera mindre delar (t.ex. 5 eller 10 delar) och tränar modellerna på olika kombinationer av dessa delar. Sedan utvärderas modellernas prestanda på de delar som inte användes för träning. Genom att jämföra resultaten från korsvalideringen kan Julia få en uppfattning om vilken modell som fungerar bäst över olika delar av datan. Detta ger en mer robust bedömning än att bara använda träningsdata eller testdata enskilt.

1. Vad är ”regressionsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden?

Regressionsproblem handlar om att förutsäga kvantitativa variabler baserat på andra variabler. Exempel på modeller inkluderar linjär, logistisk och polynomisk regression, samt Ridge och Lasso regression för att hantera överanpassning. Tillämpningsområden inkluderar förutsägelse av huspriser, köpbeteenden, växttillväxt, aktiepriser och framtida försäljning. Regressionsmodeller används över olika branscher för att förstå och förutsäga kvantitativa fenomen.

1. Hur kan du tolka RMSE och vad används det till: 𝑅𝑀𝑆𝐸 = √ 1 𝑛 ∑(𝑦𝑖 − 𝑦̂𝑖 ) 2

Root Mean Squared Error (RMSE) measures the distance between data points and the regression line. It's utilized to find the most optimal regression line by aiming for a lower RMSE, indicating closer proximity of data points to the line. Therefore, a low RMSE signifies good performance, while a high RMSE indicates poorer performance.

1. Vad är ”klassificieringsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden? Vad är en ”Confusion Matrix”?

Classification problems arise when the dependent variable y can take on one of a predetermined number of classes.

Some models that can be used for classification problems include RandomForestClassifier, KNeighborsClassifier, and LinearSVC.

A Confusion Matrix is a matrix that summarizes how well a model performs. It displays True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). Through these metrics, one can calculate the model's recall, accuracy, precision, F1-score, and thus gauge its performance.

1. Vad är K-means modellen för något? Ge ett exempel på vad det kan tillämpas på.

K-means clustering är en osuperviserad inlärningsalgoritm som används för att dela upp en datamängd i kluster baserat på likheter mellan observationer.

K-means: Delar upp objekt i flera kluster, där varje kluster har liknande egenskaper. Antalet kluster anges med K. Exempel på tillämpning: Om vi har kunddata kan vi använda K-means för att skapa grupper av liknande kunder. Sedan kan vi rikta olika typer av marknadsföring till varje grupp.

1. Förklara (gärna med ett exempel): Ordinal encoding, one-hot encoding, dummy variable encoding. Se mappen ”l8” på GitHub om du behöver repetition.

Ordinal Encoding ger varje kategori ett nummer baserat på dess ordning. One-Hot Encoding skapar binära kolumner för varje kategori. Dummy Variable Encoding gör samma sak som One-Hot Encoding men undviker att skapa en extra kolumn för att minska riskerna för problem i modellen.

1. Göran påstår att datan antingen är ”ordinal” eller ”nominal”. Julia säger att detta måste tolkas. Hon ger ett exempel med att färger såsom {röd, grön, blå} generellt sett inte har någon inbördes ordning (nominal) men om du har en röd skjorta så är du vackrast på festen (ordinal) – vem har rätt?

Jag håller med Julia. Färgerna i sig själva har ingen inbördes ordning och klassificeras därför som nominala. Men när vi säger att någon är "den vackraste på festen" baserat på färgen på deras skjorta, så innebär det en implicit rangordning av färgerna baserat på deras attraktivitet i den givna kontexten. Därför kan det tolkas som en ordinal variabel.

1. Vad är Streamlit för något och vad kan det användas till?

Streamlit is an open-source framework designed specifically for creating data applications in Python. It's particularly useful for Machine Learning and Data Science teams as it simplifies the process of building interactive web applications for data analysis and model deployment. With Streamlit, teams can quickly prototype, visualize data, and share their findings with others, enhancing collaboration and productivity in data-driven projects.

# Självutvärdering

1. Utmaningar du haft under arbetet samt hur du hanterat dem.

Svårigheterna jag stötte på var rapporten. Denna kurs var inte min favorit kurs, jag kände svårigheter med kursen och tidsbristen på grund av min sjukdom det senaste perioden, jag var stressad och mitt psykiska tillstånd var inte bra.

1. Vilket betyg du anser att du skall ha och varför.

Jag nöjer mig med ett G . Denna kursen var jätte svårt och jag gjorde så gott jag kunde.

1. Något du vill lyfta fram till Antonio?

Jag hoppas att läraren tar hänsyn till att jag aldrig har läst IT kurser på svenska så språket är ett stor hinder för mig. Trots allt så kämpar jag dag och natt för att klara kurserna.